



Mineure CalHau1

Bibliothèque BLAS

Stéphane Vialle



Stephane.Vialle@centralesupelec.fr
<http://www.metz.supelec.fr/~vialle>

Bibliothèques BLAS

- 1 – Les bibliothèques BLAS
- 2 – La fonction *cblas_dgemm*
- 3 – Espace de stockage pour *cblas_dgemm*

1 - Les bibliothèques « BLAS »

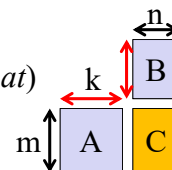
BLAS : Basic Linear Algebra Subprograms

- Ensemble de fonctions d'algèbre linéaire
- Prototypes publiés en 1979
https://fr.wikipedia.org/wiki/Basic_Linear_Algebra_Subprograms
- 3 niveaux de BLAS :
 - niveau 1 : opérations sur les vecteurs
 - niveau 2 : opérations de type matrice – vecteur
 - niveau 3 : opérations de type matrice – matrice
- **UNE API standardisée et DES implantations**
 - implantations open-sources
ex : OpenBLAS, ATLAS (très efficaces)
 - implantations propriétaires ciblées sur un type de matériel
ex : bibliothèque MKL d'Intel
 - **installer en recompilant sur la machine cible**

2 - La Fonction `cblas_dgemm`

- BLAS niveau 3
- Fonction de produit de matrices denses
- Travaille sur des *double* (`cblas_sgemm` pour les *float*)

$$C = \alpha \cdot op(A) \times op(B) + \beta \cdot C$$



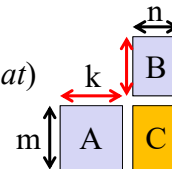
```
void cblas_dgemm (
  const CBLAS_LAYOUT layout,
  const CBLAS_TRANSPOSE transa,
  const CBLAS_TRANSPOSE transb,
  const int m, const int n, const int k,
  const double alpha,
  const double *a, const int lda,
  const double *b, const int ldb,
  const double beta,
  double *c, const int ldc)
```

Stockage en
row major ou
column major

2 - La Fonction **cblas_dgemm**

- BLAS niveau 3
- Fonction de produit de matrices denses
- Travaille sur des *double* (**cblas_sgemm** pour les *float*)

$$C = \alpha.op(A) \times op(B) + \beta.C$$



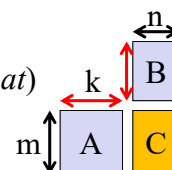
```
void cblas_dgemm (
    const CBLAS_LAYOUT Layout,
    const CBLAS_TRANSPOSE transa,
    const CBLAS_TRANSPOSE transb,
    const int m, const int n, const int k,
    const double alpha,
    const double *a, const int lda,
    const double *b, const int ldb,
    const double beta,
    double *c, const int ldc)
```

Transposition
« au vol » (ou
pas) de A et B

2 - La Fonction **cblas_dgemm**

- BLAS niveau 3
- Fonction de produit de matrices denses
- Travaille sur des *double* (**cblas_sgemm** pour les *float*)

$$C = \alpha.op(A) \times op(B) + \beta.C$$



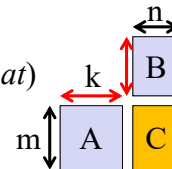
```
void cblas_dgemm (
    const CBLAS_LAYOUT Layout,
    const CBLAS_TRANSPOSE transa,
    const CBLAS_TRANSPOSE transb,
    const int m, const int n, const int k,
    const double alpha,
    const double *a, const int lda,
    const double *b, const int ldb,
    const double beta,
    double *c, const int ldc)
```

Tailles des
matrices

2 - La Fonction `cblas_dgemm`

- BLAS niveau 3
- Fonction de produit de matrices denses
- Travaille sur des *double* (`cblas_sgemm` pour les *float*)

$$C = \alpha \cdot op(A) \times op(B) + \beta \cdot C$$



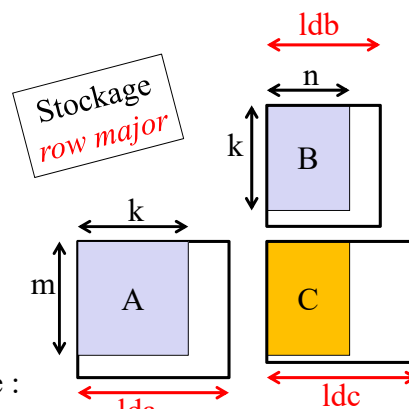
```
void cblas_dgemm (
    const CBLAS_LAYOUT Layout,
    const CBLAS_TRANSPOSE transa,
    const CBLAS_TRANSPOSE transb,
    const int m, const int n, const int k,
    const double alpha,
    const double *a, const int lda,
    const double *b, const int ldb,
    const double beta,
    double *c, const int ldc)
```

lda, ldb, ldc : tailles des zones de stockage (voir plus loin)

3 – Espace de stockage pour `cblas_dgemm`

Les espaces de stockage peuvent être plus grand que les matrices :

- Réutilisation d'un espace alloué précédemment
- Espace de stockage dimensionné au maximum
- Calcul sur des sous-matrices
- ...



La fonction `cblas_dgemm` doit connaître :

- le nombre de colonnes de la matrice
 - le nombre de colonnes de l'espace de stockage
- Pour savoir combien d'espace sauter entre deux lignes

Ex : en *row major* sans transposer A et B :

il faut sauter $lda - k$, avec : $lda \geq k$, $ldb \geq n$, $ldc \geq n$

Bibliothèques BLAS

FIN